

Continuous-time hybridization expansion quantum impurity solver for multi-orbital systems with complex hybridizations[☆]

Hiroshi Shinaoka^{a,b,*}, Emanuel Gull^c, Philipp Werner^d

^a Department of Physics, Saitama University, Saitama 338-8570, Japan

^b Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland

^c University of Michigan, Ann Arbor, MI 48109, USA

^d Department of Physics, University of Fribourg, 1700 Fribourg, Switzerland

We describe an open-source implementation of the continuous-time hybridization-expansion quantum Monte Carlo method for impurity models with general instantaneous two-body interactions and complex hybridization functions. The code is built on an updated version of the core libraries of ALPS (Applications and Libraries for Physics Simulations) [ALPSCore libraries].

Program summary

Program title: ALPSCore CT-HYB

Program Files doi: "<http://dx.doi.org/10.17632/dyhxx6g4md.1>"

Licensing provisions: GPLv3

Programming language: C++, MPI for parallelization.

Nature of problem: Quantum impurity problem

Solution method: Continuous-time hybridization-expansion quantum Monte Carlo

External routines/libraries: ALPSCore libraries, Eigen3, Boost.

1. Introduction

In condensed matter physics, dynamical mean-field theory (DMFT) [1] is a widely used tool for the study of strongly correlated electron systems. In a DMFT calculation, a correlated lattice model is mapped to an impurity problem whose bath degrees of freedom are self-consistently determined. Although the approximation was originally proposed for the single-band Hubbard model, the DMFT formalism can be extended to multi-orbital systems and cluster-type impurities [2]. Furthermore, DMFT can be combined with density functional theory based ab-initio calculations, to describe strongly correlated materials such as transition metal oxides [3]. In such realistic applications, one may have to treat local Coulomb interactions with non-density-density terms. Furthermore, in simulations of 4d and 5d transition metal oxides, spin-orbit coupling gives rise to complex hybridization functions [4]. In the presence of multiple local degrees of freedom or complex hybridization

functions, the solution of the quantum impurity problem becomes a numerically costly task. For the DMFT self-consistency loop, we only have to compute single-particle quantities such as the self-energy. However, it may be desirable to compute also higher-order correlation functions, to get access to spin-orbital susceptibilities, or in the context of diagrammatic extensions of the DMFT formalism [5,6].

Continuous-time Monte Carlo is a general framework to solve a quantum model in a numerically exact way by sampling a series expansion of the partition function. There are two complementary algorithms for quantum impurity problems with general interactions, called CT-INT (continuous-time interaction expansion) and CT-HYB (continuous-time hybridization expansion). The former is based on the expansion of the partition function with respect to the local interaction [7], while the latter is based on an expansion in the hybridization between the bath and the impurity [8]. CT-HYB is particularly efficient in the strongly correlated regime [9], and it is widely used for both model and material simulations.

The available open-source implementations of CT-HYB [10–12] support neither complex hybridization functions nor the measurement of the two-particle Green's function for general multi-orbital models. In this paper, we describe a state-of-the-art implementation of CT-HYB for multiple orbitals and complex hybridization functions. The code provides measurements of various observables

[☆] This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author at: Department of Physics, Saitama University, Saitama 338-8570, Japan.

E-mail address: shinaoka@mail.saitama-u.ac.jp (H. Shinaoka).

and correlation functions relevant for DMFT calculations, including the single-particle Green's function, density-density correlations, and four-time/two-time two-particle Green's functions. The measurements are performed by worm sampling [13–16], to avoid ergodicity problems.

The remainder of this paper is organized as follows. In Section 2, we introduce a general quantum impurity model. In Section 3, we describe CT-HYB and worm sampling. Section 4 describes the evaluation of the trace over local degrees of freedom, while Section 5 explains the possibility to change the single-particle basis. The installation and usage is detailed in Section 6 and Section 7 provides some examples of simulation results for a three-orbital model. Finally, we summarize the paper in Section 8.

2. General impurity model

We consider a general multi-orbital impurity model defined by the Hamiltonian

$$\mathcal{H}_{\text{imp}} = \mathcal{H}_{\text{loc}} + \mathcal{H}_{\text{bath}} + \mathcal{H}_{\text{hyb}}, \quad (1)$$

where

$$\mathcal{H}_{\text{loc}} = \sum_{ab} t_{ab} c_a^\dagger c_b + \sum_{abcd} V_{abcd} c_a^\dagger c_b^\dagger c_c c_d, \quad (2)$$

$$\mathcal{H}_{\text{bath}} = \sum_{\alpha} \epsilon_{\alpha} d_{\alpha}^{\dagger} d_{\alpha}, \quad (3)$$

$$\mathcal{H}_{\text{hyb}} = \sum_{\alpha, b} V_{\alpha b} d_{\alpha}^{\dagger} c_b + \text{h.c.} \quad (4)$$

The indices a and b denote the internal degrees of freedom of the impurity, which we call *flavors*, e.g., a composite index of spin and orbital, while α denotes those of the bath. The chemical potential is absorbed into t_{ab} .

For the Monte Carlo simulation, we switch to an action formulation and trace out the bath degrees of freedom. In the action

$$\mathcal{S}_{\text{imp}} = \int_0^{\beta} d\tau \mathcal{H}_{\text{loc}}(\tau) + \int_0^{\beta} d\tau d\tau' \sum_{ab} c_a^{\dagger}(\tau) \Delta_{ab}(\tau - \tau') c_b(\tau') \quad (5)$$

all relevant local information of $\mathcal{H}_{\text{bath}}$ and \mathcal{H}_{hyb} is encoded in the hybridization function defined by

$$\Delta_{ab}(i\omega_n) = \sum_{\alpha} \frac{V_{\alpha a}^* V_{\alpha b}}{i\omega_n - \epsilon_{\alpha}}. \quad (6)$$

Here, β is the inverse temperature and $\omega_n = (2n + 1)\pi/\beta$ a Matsubara frequency. The Fourier transformation to imaginary time can be written as

$$\begin{aligned} \Delta_{ab}(\tau) &= \frac{1}{\beta} \sum_{n=-\infty}^{\infty} e^{-i\omega_n \tau} \Delta_{ab}(i\omega_n) \\ &= \frac{1}{\beta} \sum_{n=0}^{\infty} e^{-i\omega_n \tau} \{ \Delta_{ab}(i\omega_n) + \Delta_{ba}^*(i\omega_n) \}. \end{aligned} \quad (7)$$

The hybridization function is fermionic and hence β -antiperiodic, $\Delta_{ab}(\tau + \beta) = -\Delta_{ab}(\tau)$, with discontinuities at $\tau = n\beta$ (n integer). For the simulations, we define $\Delta(\tau)$ on the interval $[0, \beta]$ in a continuous fashion. In the original literature [8,17], there is an alternative notation for the hybridization function, F_{ab} , which is related to Δ_{ab} by

$$F_{ab}(\tau) = -\Delta_{ba}(\beta - \tau) \quad (0 \leq \tau \leq \beta). \quad (8)$$

Our impurity solver takes $\Delta(\tau)$ as an input in addition to the transfer matrix t_{ab} and the Coulomb tensor U_{abcd} . Note that the

hybridization function can have diagonal and offdiagonal components, and that their values may be complex. The latter property is for example essential for the simulation of models with spin-orbit coupling [4]. The different components must satisfy the relation $\Delta_{ab}(\tau) = \Delta_{ba}^*(\tau)$.

3. Hybridization expansion and worm sampling

3.1. Expansion of the partition function

For a given inverse temperature β , the partition function Z of the impurity model is expanded in powers of the hybridization as

$$\begin{aligned} Z &\equiv \langle e^{-\beta \mathcal{H}} \rangle \\ &\propto \sum_{n=0}^{\infty} \frac{1}{n!^2} \sum_{a_1, \dots, a_n} \sum_{a'_1, \dots, a'_n} \int_0^{\beta} d\tau_1 d\tau'_1 \cdots \int_0^{\beta} d\tau_n d\tau'_n \\ &\quad \times \text{Tr}_{\text{loc}} \left[e^{-\beta \mathcal{H}_{\text{loc}}} T_{\tau} c_{a_1}(\tau_1) c_{a'_1}^{\dagger}(\tau'_1) \cdots c_{a_n}(\tau_n) c_{a'_n}^{\dagger}(\tau'_n) \right] \\ &\quad \times \det \mathbf{M}^{-1}, \end{aligned} \quad (9)$$

where n is the order of the expansion, $c_a^{\dagger}(\tau) = e^{\tau \mathcal{H}_{\text{loc}}} c_a^{\dagger} e^{-\tau \mathcal{H}_{\text{loc}}}$ and $c_a(\tau) = e^{\tau \mathcal{H}_{\text{loc}}} c_a e^{-\tau \mathcal{H}_{\text{loc}}}$. The matrix elements of $(\mathbf{M}^{-1})_{ij}$ are given by the hybridization function,

$$(\mathbf{M}^{-1})_{ij} = \Delta_{a'_i, a_j}(\tau'_i - \tau_j). \quad (10)$$

Eq. (9) can be recast into

$$\begin{aligned} Z &\propto \sum_{n=0}^{\infty} \sum_{a_1, \dots, a_n} \sum_{a'_1, \dots, a'_n} \int_0^{\tau_2} d\tau_1 \int_0^{\tau'_2} d\tau'_1 \cdots \int_0^{\beta} d\tau_n \int_0^{\beta} d\tau'_n \\ &\quad \times \text{Tr}_{\text{loc}} \left[e^{-(\beta - \tilde{\tau}_{2n}) \mathcal{H}_{\text{loc}}} O_{2n} \cdots e^{-(\tilde{\tau}_2 - \tilde{\tau}_1) \mathcal{H}_{\text{loc}}} O_1 e^{-\tilde{\tau}_1 \mathcal{H}_{\text{loc}}} \right] \\ &\quad \times (-1)^{P_{\text{trace}}} \times \det \mathbf{M}^{-1}, \end{aligned} \quad (11)$$

where $0 \leq \tau_1 < \tau_2 < \cdots < \tau_n < \beta$, $0 \leq \tau'_1 < \tau'_2 < \cdots < \tau'_n < \beta$, and $\{O_1, \dots, O_{2n}\}$ is a time-ordered set of the impurity creation and annihilation operators. P_{trace} is the permutation of time ordering from $\{c_{a_1}(\tau_1), c_{a'_1}^{\dagger}(\tau'_1), \dots, c_{a_n}(\tau_n), c_{a'_n}^{\dagger}(\tau'_n)\}$ to $\{O_1, \dots, O_{2n}\}$. The set $\{\tilde{\tau}_1, \dots, \tilde{\tau}_{2n}\}$ is a time-ordered set of $\{\tau_1, \dots, \tau_n, \tau'_1, \dots, \tau'_n\}$. In the Monte Carlo simulations, we perform an importance sampling of Z using the configurations $c = \{\tau_1, \tau'_1, \dots, \tau_n, \tau'_n; a_1, a'_1, \dots, a_n, a'_n\}$ and the Metropolis algorithm. Since the weight

$$\begin{aligned} w(c) &= \text{Tr}_{\text{loc}} \left[e^{-(\beta - \tilde{\tau}_{2n}) \mathcal{H}_{\text{loc}}} O_{2n} \cdots \right. \\ &\quad \left. e^{-(\tilde{\tau}_2 - \tilde{\tau}_1) \mathcal{H}_{\text{loc}}} O_1 e^{-\tilde{\tau}_1 \mathcal{H}_{\text{loc}}} \right] (-1)^{P_{\text{trace}}} \det \mathbf{M}^{-1} (d\tau)^{2n} \end{aligned} \quad (12)$$

can be complex, we consider the absolute value $|w(c)|$ in the sampling. That is, we sample an unphysical partition function

$$\bar{Z} \equiv \sum_c |w(c)|. \quad (13)$$

The ratio \bar{Z}/Z appears as a reweighing factor in the estimators of observables.

3.2. Worm sampling of the Green's function

We now briefly describe how to measure the expectation values of correlation functions. For instance, the single-particle Green's function is defined as

$$G_{ij}(\tau - \tau') = - \frac{\text{Tr}[T_{\tau} e^{-\mathcal{S}_{\text{imp}}} c_i(\tau) c_j^{\dagger}(\tau')]}{Z}, \quad (14)$$

where $\tau, \tau' \in [0, \beta]$. This quantity is the most important observable for DMFT calculations.

Similarly to Eq. (9), the numerator of Eq. (14) is expanded as

$$\begin{aligned} \text{Tr}[T_\tau e^{-S_{\text{imp}}} c_i(\tau) c_j^\dagger(\tau')] &= Z_B \sum_{n=0}^{\infty} \sum_{a_1, \dots, a_n} \sum_{a'_1, \dots, a'_n} \\ &\int_0^{\tau_2} d\tau_1 \int_0^{\tau'_2} d\tau'_1 \dots \int_0^{\beta} d\tau_n \int_0^{\beta} d\tau'_n \\ &\times \text{Tr}_{\text{loc}} \left[e^{-(\beta - \tilde{\tau}_{2n+2})H_{\text{loc}}} O_{2n+2} \dots \right. \\ &\quad \left. \dots e^{-(\tilde{\tau}_2 - \tilde{\tau}_1)H_{\text{loc}}} O_1 e^{-\tilde{\tau}_1 H_{\text{loc}}} \right] \\ &\times (-1)^{P_{\text{trace}}} \det \mathbf{M}^{-1}, \end{aligned} \quad (15)$$

where $\{O_1, \dots, O_{2n+2}\}$ is a time-ordered set of the impurity creation and annihilation operators, and those of the Green's function. P_{trace} is the permutation of time ordering from $\{c_{a_1}(\tau_1), c_{a'_1}^\dagger(\tau'_1), \dots, c_{a_n}(\tau_n), c_{a'_n}^\dagger(\tau'_n), c_i(\tau), c_j^\dagger(\tau')\}$ to $\{O_1, \dots, O_{2n+2}\}$. The set $\{\tilde{\tau}_1, \dots, \tilde{\tau}_{2n+2}\}$ is a time-ordered set of $\{\tau_1, \dots, \tau_n, \tau'_1, \dots, \tau'_n, \tau, \tau'\}$.

To construct the estimator of Eq. (14), we enlarge the configuration space as [13–15]

$$C = C_Z \oplus C_{G(1)}, \quad (16)$$

where a configuration c_A in the Green's function space C_A is defined as $c_A = \{\tau_1, \tau'_1, \dots, \tau_n, \tau'_n; a_1, a'_1, \dots, a_n, a'_n; c_i(\tau), c_j^\dagger(\tau')\}$. The contribution of c_A to the expectation value reads

$$\begin{aligned} w(c_A) &= \text{Tr}_{\text{loc}} \left[e^{-(\beta - \tilde{\tau}_{2n+2})H_{\text{loc}}} O_{2n+2} \dots \right. \\ &\quad \left. \dots e^{-(\tilde{\tau}_2 - \tilde{\tau}_1)H_{\text{loc}}} O_1 e^{-\tilde{\tau}_1 H_{\text{loc}}} \right] (-1)^{P_{\text{trace}}} \det \mathbf{M}^{-1} (d\tau)^{2n+2}, \end{aligned} \quad (17)$$

up to a normalization factor. We sample in both configuration spaces according to weight $|w(c)|$ and $|w(c_A)|$, respectively. In practice, we switch from C_Z to $C_{G(1)}$ by inserting a “worm” $\{c_i(\tau), c_j^\dagger(\tau')\}$. We return back to C_Z by removing a worm.

The total partition function \bar{W} is defined as

$$\bar{W} = \bar{Z} + \eta \bar{Z}_{G(1)} \quad (18)$$

with

$$\bar{Z}_{G(1)} = \int_0^\beta d\tau d\tau' \left| \text{Tr}[T_\tau e^{-S_{\text{imp}}} c(\tau) c^\dagger(\tau')] \right|. \quad (19)$$

The overline means that we take the absolute values of the contributions of diagrams. The coefficient η (> 0) was introduced so that the simulation spends almost the same number of Monte Carlo steps in both spaces. It appears as an additional factor in the weight $|w(c_A)|$. The parameter η is adjusted during the thermalization process using the Wang–Landau algorithm [18,19]. We refer the reader to Appendix A for more details.

Considering the anti-periodicity of the Green's function with respect to β , the estimator reads

$$G(\Delta\tau) = \frac{\beta^{-1} \eta^{-1} N_{G(1)} \langle \text{sign} \tilde{\delta}(\Delta\tau - (\tau - \tau')) \rangle_{G(1)}}{N_Z \langle \text{sign} \rangle_Z}, \quad (20)$$

for $0 < \Delta\tau < \beta$. To simplify the notation, we introduced

$$\tilde{\delta}(\tau) = \begin{cases} \delta(\tau) & 0 \leq \tau < \beta \\ -\delta(\tau + \beta) & -\beta \leq \tau < 0. \end{cases}$$

The symbols “sign” in the numerator and the denominator denote $w(c_{G(1)})/|w(c_{G(1)})|$ and $w(c_Z)/|w(c_Z)|$, respectively. N_Z and $N_{G(1)}$ are the number of Monte Carlo steps spent in C_Z and $C_{G(1)}$, respectively. The brackets $\langle \dots \rangle_Z$ and $\langle \dots \rangle_{G(1)}$ denote the Monte Carlo average in C_Z and $C_{G(1)}$, respectively. The factor β^{-1} comes from the extra degree of freedom τ' in the sampling in $C_{G(1)}$.

In general, $w(c_Z)/|w(c_Z)|$ is a complex number, but the expectation value $\langle \text{sign} \rangle_Z$ is real, because the partition function Z is real.

The importance sampling works efficiently as long as $\langle \text{sign} \rangle_Z$ is not too small ($\text{sign} \gtrsim 0.1$).

In practice, instead of using the imaginary-time estimator, Eq. (20), we expand the Green's function in the Legendre polynomials defined on the interval $[0, \beta]$ [20] as

$$G_{ab}(\tau) = \sum_{l \geq 0} \frac{\sqrt{2l+1}}{\beta} P_l[x(\tau)] G_l^{ab}, \quad (21)$$

$$G_l^{ab} = \sqrt{2l+1} \int_0^\beta d\tau P_l[x(\tau)] G(\tau), \quad (22)$$

where $x(\tau) = 2\tau/\beta - 1$ and $P_l(x)$ is the l th Legendre polynomial defined on the interval $[-1, 1]$. In the Legendre basis, the estimator reads

$$G_l^{ab} = \frac{\beta^{-1} \eta^{-1} N_{G(1)} \sqrt{2l+1} \langle \text{sign} \tilde{P}_l(x(\tau - \tau')) \rangle_{G(1)}}{N_Z \langle \text{sign} \rangle_Z}, \quad (23)$$

where

$$\tilde{P}_l[x(\delta\tau)] = \begin{cases} P_l[x(\delta\tau)], & (\delta\tau > 0), \\ -P_l[x(\delta\tau + \beta)], & (\delta\tau < 0). \end{cases} \quad (24)$$

The cutoff N_l is a simulation parameter. Typical values are $N_l = 50 - 100$.

At this point, it is worth pointing out a practical limitation of this measurement. In the estimator (23), we measure the Green's function only at one time difference and a pair of flavors at each measurement step. Thus, the sampling by this estimator is less efficient than the measurement by removing hybridization lines from a configuration in C_Z as in the conventional CT-HYB [8].

To improve the statistics, we generate multiple samples from a configuration in $C_{G(1)}$ by reconnecting hybridization lines at each worm measurement step without reevaluating the trace. We refer the reader to Appendix C for more details.

3.3. Worm measurement of other quantities

We also provide worm measurements of the following correlation functions, which play an important role in DMFT calculations:

- Two-particle Green's function

$$\begin{aligned} G_{abcd}^{(2)}(\tau_1, \tau_2, \tau_3, \tau_4) &= \frac{\text{Tr}[T_\tau e^{-S_{\text{imp}}} c_a(\tau_1) c_b^\dagger(\tau_2) c_c(\tau_3) c_d^\dagger(\tau_4)]}{Z}, \end{aligned} \quad (25)$$

- Equal-time single-particle Green's function [16]

$$\frac{\text{Tr}[T_\tau e^{-S_{\text{imp}}} c_a^\dagger(\tau) c_b(\tau)]}{Z}, \quad (26)$$

- Two-time two-particle Green's function

$$\frac{\text{Tr}[T_\tau e^{-S_{\text{imp}}} c_a^\dagger(\tau) c_b(\tau) c_c^\dagger(\tau') c_d(\tau')]}{Z}. \quad (27)$$

We measure the two-particle Green's function and the two-time two-particle Green's function in the Legendre basis.

The extended configuration space for the measurement of multiple observable reads

$$\bar{W} = \bar{Z} + \eta \bar{Z}_{G(1)} + \eta_{O_1} \bar{Z}_{O_1} + \dots + \eta_{O_N} \bar{Z}_{O_N}, \quad (28)$$

where O_1, \dots, O_N are the observables measured by worm sampling. The coefficient $\eta, \eta_{O_1}, \dots, \eta_{O_N}$ are adjusted so that the simulation spends an equal number of steps in each subspace.

Worm insertion/removal updates result in transitions between \bar{Z} and the worm subspaces \bar{Z}_{O_i} . We also perform direct transitions between the worm spaces of different observables such as the equal-time single-particle Green's function and the two-time two-particle Green's function to reduce autocorrelation times.

4. Evaluation of the trace over the local Hilbert space

4.1. Construction of eigenbasis

We briefly describe how we evaluate Eq. (12), which is the heart of the CT-HYB code. The size of the local Hilbert space grows exponentially with the number of orbitals or sites in the impurity. The computational cost is greatly reduced by partitioning the local Hilbert space into sectors such that \mathcal{H}_{loc} is a block diagonal matrix and the block matrices of creation and annihilation operators have only a single nonzero block for each row [21].

After partitioning the Hilbert space into sectors, we construct an eigenbasis for each sector to rewrite the trace as

$$\begin{aligned} \text{Tr}_{\text{loc}} \left[e^{-(\beta - \tau_{2n})H_{\text{loc}}} O_{2n} \dots e^{-(\tau_2 - \tau_1)H_{\text{loc}}} O_1 e^{-\tau_1 H_{\text{loc}}} \right] \\ = \sum_s \text{Tr} \left\{ \mathbf{E}^{s, s_{2n}}(\beta - \tau_{2n}) \mathbf{O}_{2n}^{s_{2n}, s_{2n-1}} \dots \right. \\ \left. \dots \mathbf{E}^{s_1}(\tau_2 - \tau_1) \mathbf{O}_1^{s_1, s} \mathbf{E}^s(\tau_1) \right\}, \end{aligned} \quad (29)$$

where $\mathbf{E}^s(\tau)$ is the diagonal matrix $\{e^{-\epsilon_n^s \tau}\}$ (ϵ_n^s is the n th eigenvalue of the sector s). $\mathbf{O}^{s, s'}$ is the matrix representation of an annihilation d or a creation operator d^\dagger . For a given sector s' , there is only one non-zero block matrix $\mathbf{O}^{s, s'}$ (s is determined uniquely).

The partitioning of the Hilbert space may be done by exploiting conserved quantum numbers that commute with \mathcal{H}_{loc} . Examples include the total electron number as well as the special conserved quantities which commute with the Slater-Kanamori Hamiltonian [22]. Recently, it was shown that it is possible to partition the local Hilbert space without prior knowledge of quantum numbers for CT-HYB [11, 23]. This is done by looking at non-zero elements of the matrix representations of \mathcal{H}_{loc} and c_d^\dagger in the occupation basis. Our code provides the same functionality based on an efficient cluster identification algorithm. We refer the readers to Appendix B for more technical details.

4.2. Sliding-window update

Monte Carlo updates consist of elementary updates such as the insertion/removal of a pair of c and c^\dagger . To compute the acceptance rate of a new configuration, we evaluate the trace using Eq. (29). This may cost $O(\beta)$ operations because the number of matrices to be multiplied increases linearly with β . We reduce the computational cost by using the sliding-window update proposed in Ref. [24]. As illustrated in Fig. 1(a), we define a narrow window in which the updates are performed on the imaginary time axis. The left and right end points are τ_L and τ_R ($\tau_R - \tau_L = \tau_{\text{win}}$). The idea is that we precompute the products of all matrices for $\tau > \tau_L$ and $\tau < \tau_R$, respectively, and store them in memory.

We define a *ket* as

$$\begin{aligned} |\tau_R, s\rangle &= \mathbf{R}(\tau_R, s) \\ &\equiv \mathbf{E}^{s_{n_R+1}, s_{n_R}}(\tau_R - \tau_{n_R}) \mathbf{O}_{n_R}^{s_{n_R}, s_{n_R-1}} \dots \\ &\dots \mathbf{E}^{s_1}(\tau_2 - \tau_1) \mathbf{O}_1^{s_1, s} \mathbf{E}^s(\tau_1), \end{aligned} \quad (30)$$

where n_R is the number of operators on the interval $(\tau_R, 0]$. We define a *bra* in a similar way as

$$\begin{aligned} \langle \tau_L, s| &= \mathbf{L}(\tau_L, s) \\ &\equiv \mathbf{E}^{s, s_{2n}}(\beta - \tau_{2n}) \mathbf{O}_{2n}^{s_{2n}, s_{2n-1}} \dots \\ &\dots \mathbf{O}_{n_L}^{s_{n_L}, s_{n_L-1}} \mathbf{E}^{s_{n_L-1}}(\tau_{n_L} - \tau_L), \end{aligned} \quad (31)$$

where n_L is the index of the operator with the smallest imaginary time on the interval $(\beta, \tau_L]$. Then, the trace is rewritten as

$$\sum_s \langle \tau_L, s | \mathbf{Q}(\tau_L, \tau_R) | \tau_R, s \rangle, \quad (32)$$

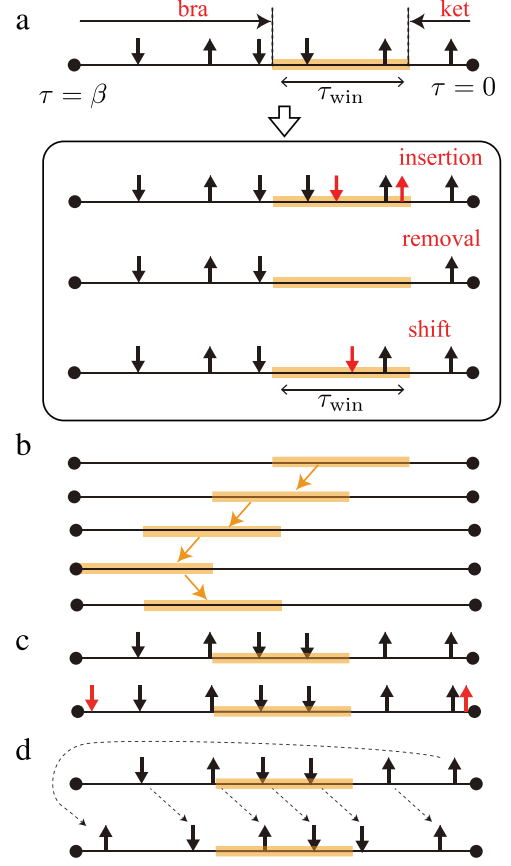


Fig. 1. (Color online) (a) Elementary updates in the sliding window. (b) Sequential sweep of the window on the imaginary time axis. (c) Insertion of creation and annihilation operators crossing the boundary at $\beta = 0$. (d) Global update in which all operators are shifted on the imaginary time axis.

where $\mathbf{Q}(\tau_L, \tau_R)$ is the product of matrices on the interval $(\tau_L, \tau_R]$. We do not have to recompute the bra and ket as long as updates are performed within the window.

We propose a few elementary updates, whose number is proportional to that of flavors, at each position of the window. After that, we move the window to the next position with a finite overlap with the previous position as illustrated in Fig. 1(b). We update the bra and ket by applying creation, annihilation operators and time-evolution operators, or by loading cached data from memory. Note that we do not have to calculate the bra and ket from scratch thanks to the sequential move of the window on the imaginary-time axis. The procedure is repeated by moving the window sequentially back and forth on the whole interval $[0, \beta]$.

A reasonable value of the window width τ_{win} is automatically estimated during the thermalization process. We choose $\tau_{\text{win}} = \beta / \langle n \rangle_{\text{MC}}$, where $\langle n \rangle_{\text{MC}}$ is the Monte Carlo average of the perturbation order.

In the sliding-window update, we never insert/remove a pair of creation and annihilation operators which cross the boundary at $\beta \equiv 0$ as illustrated in Fig. 1(c). To avoid this problem, we also perform a global update in which all the operators are shifted on the imaginary-time axis by a fixed random time $\Delta\tau \in [0, \beta]$ [see Fig. 1(d)]. This update is proposed each time the window has completed a back-and-forth run on the whole interval. If this update is accepted, the bra and the ket are recomputed from scratch, which costs $O(\beta)$. However, its computational cost is typically smaller than that of the sliding-window update because it is performed less frequently.

4.3. Restricting the trace to the active space

To further reduce the computational cost, we offer several options of eliminating high-energy states which do not contribute to low-temperature physics. The first one, called “inner-outer-state cutoff”, was originally introduced in Ref. [21]. Here, we simplify does not include the eigenstates of \mathcal{H}_{loc} whose energies are higher than a certain cutoff in the construction of the block matrices. As a consequence, the matrices appearing in Eq. (29) are reduced in size.

The second option is called “outer-state cutoff”, and was originally introduced in the Krylov algorithm [25]. In Eq. (30), the right most matrix is modified as

$$\mathbf{E}^s(\tau_1) \rightarrow \mathbf{E}^s(\tau_1)\mathbf{P}^s, \quad (33)$$

where \mathbf{P}^s is a projector to the active eigenstates in the sector s , e.g., eigenstates whose energies are lower than a certain value. We replace the left most matrix in the bra Eq. (31) in a similar way as well. If we keep only a few eigenstates in the projector, the cost will be reduced from $O(N_H^3)$ to $O(N_H^2)$. This may lead to a substantial speed-up when the linear dimensions of the block matrices are larger than ≈ 20 .

In Ref. [25], it was demonstrated that at low temperatures, the approximate result converges to the exact result even when only the ground states of \mathcal{H}_{loc} are kept in the projector. This approximation is safer than the “inner-outer-state cutoff” because all eigenstates remain accessible during the imaginary-time evolution.

5. Single-particle basis transformation

In general, the average sign $\langle \text{sign} \rangle_Z$ depends on the choice of the single-particle basis used for expanding the partition function with respect to the hybridization function in Eq. (11). It is practically useful to use a single-particle basis which gives a larger average sign to improve the statistics.

To this end, our CT-HYB solver provides a functionality which allows the user to perform the expansion using an arbitrary single-particle local basis. A transformed single-particle basis is defined as

$$\tilde{c}_a = \sum_b U_{ba}^* c_b, \quad (34)$$

$$c_a^\dagger = \sum_b U_{ba} c_b^\dagger, \quad (35)$$

with U_{ab} being a unitary matrix, which is an input parameter. The local Hamiltonian and the hybridization function are transformed to the new single-particle basis as described in Ref. [26,27] before a Monte Carlo simulation. The user inputs t_{ab} , V_{abcd} and Δ_{ab} in the original basis to the impurity solver which takes care of their transformation.

During the Monte Carlo simulation, we measure all the observables such as the single-particle Green's function in the transformed basis. Then, the measured results are transformed back to the original basis after the Monte Carlo simulation except for the density-density correlations.

For instance, we collect the single-particle Green's function

$$\tilde{G}_{ij}(\tau - \tau') = -\frac{\text{Tr}[T_\tau e^{-S_{\text{imp}}} \tilde{c}_i(\tau) \tilde{c}_j^\dagger(\tau')]}{Z} \quad (36)$$

during the Monte Carlo simulation. Once the simulation is done, the data are transformed back to the original basis as

$$G_{ij}(\tau - \tau') = \sum_{\tilde{i}\tilde{j}} U_{i\tilde{i}} \tilde{G}_{\tilde{i}\tilde{j}}(\tau - \tau') (U^\dagger)_{\tilde{j}j}. \quad (37)$$

This functionality allows the user to choose any arbitrary basis transformation to improve the statistics.

6. Usage

The CT-HYB code is built on an updated version of the core libraries of ALPS (Applications and Libraries for Physics Simulations libraries) [ALPScore libraries] [28], the Boost libraries, and Eigen. Eigen is a C++ template header-file-only library for linear algebra. They must be pre-installed. One needs a MPI C++ compiler which supports C++03 to build the solver. At runtime, one can choose either a complex-number solver or a real-number solver. The two solvers have exactly the same interface. The CT-HYB solver reads parameters from a text file. In the next section, we discuss several examples.

The latest version of the code is available from a public Git repository at <https://github.com/ALPScore/CT-HYB>. One can also find a more detailed description of usage in Wiki documentation pages at <https://github.com/ALPScore/CT-HYB/wiki>.

7. Example: three-orbital t_{2g} model with Slater-Kanamori interaction

7.1. Model

We consider a three-orbital model for the t_{2g} shell of d orbitals with a Slater-Kanamori interaction and spin-orbit coupling. Its Hamiltonian is given by

$$\mathcal{H} = \sum_{ij} \sum_{\sigma\sigma'} h_{i\sigma,j\sigma'}^{\text{LS}} c_{i\sigma}^\dagger c_{j\sigma'} + \frac{1}{2} \sum_{ijkl} \sum_{\sigma\sigma'} V_{ijkl} c_{i\sigma}^\dagger c_{j\sigma'}^\dagger c_{k\sigma'} c_{l\sigma}, \quad (38)$$

where $c_{i\sigma}^\dagger$ and $c_{i\sigma}$ are creation/annihilation operators of an electron at orbital i with spin σ . The first term denotes the spin-orbit coupling for the t_{2g} basis. If the states are ordered as $d_{xy\uparrow}, d_{xy\downarrow}, d_{yz\uparrow}, d_{yz\downarrow}, d_{zx\uparrow}, d_{zx\downarrow}$, its matrix elements are

$$h_{i\sigma,j\sigma'} = \frac{\zeta}{2} \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & -i \\ 0 & 0 & -1 & 0 & -i & 0 \\ 0 & -1 & 0 & 0 & i & 0 \\ 1 & 0 & 0 & 0 & 0 & -i \\ 0 & i & -i & 0 & 0 & 0 \\ i & 0 & 0 & i & 0 & 0 \end{pmatrix}, \quad (39)$$

where ζ (>0) is the amplitude of the spin-orbit coupling. The none-zero elements of the Coulomb tensor are given by $V_{iiii} = U$, $V_{ijji} = U - 2J$, $V_{ijij} = J$, $V_{iiji} = J$. The chemical potential is chosen such that the system is close to half filling: $\mu = \frac{5}{2}U - 5J$.

The bath consists of three non-interacting orbitals. The intra-bath Hamiltonian reads

$$\mathcal{H}_{\text{bath}} = \sum_{i \neq j} t' d_{i\sigma}^\dagger d_{j\sigma} = \sum_{ij} h_{ij}^{\text{bath}} d_{i\sigma}^\dagger d_{j\sigma}, \quad (40)$$

where $t' \neq 0$ gives rise to off-diagonal elements of the hybridization function. The hybridization term reads

$$\mathcal{H}_{\text{hyb}} = \lambda \sum_i (c_{i\sigma}^\dagger d_{i\sigma} + d_{i\sigma}^\dagger c_{i\sigma}), \quad (41)$$

where we take the coupling $\lambda = 1$.

From Eqs. (40) and (41), we obtain the hybridization function

$$\Delta_{i\sigma j\sigma'}(\tau) = -\frac{\lambda^2 \delta_{\sigma\sigma'}}{\beta} \sum_{n=-\infty}^{\infty} \frac{e^{-i\omega_n \tau}}{i\omega_n I - h^{\text{bath}}}, \quad (42)$$

where I is an identity matrix.

7.2. Diagonal hybridization function

We first solve the model with a diagonal hybridization function, i.e., for $t' = 0$. We take $\zeta = 1, U = 10, J/U = 1/4, \beta = 40$. In Fig. 2, we compare the single-particle Green's function $G(\tau)$ measured by the worm sampling with numerically exact results computed by pomerol [29]. The average sign is about 0.95. The number of Legendre polynomials is $N_l = 50$. The perturbation expansion was performed in the original t_{2g} basis. One can see that our code can measure both the diagonal (G_{00}) and off-diagonal elements (G_{02}, G_{05}). Using the conventional sampling method, it would not be possible to measure the off-diagonal elements of $G(\tau)$ in the case of a diagonal hybridization function.

7.3. Off-diagonal hybridization function

Next, we solve the model for $t' = 0.2, \zeta = 1, U = 10, J/U = 1/4, \beta = 10$. In Fig. 3, we compare the computed results with exact results. The number of Legendre polynomials is $N_l = 30$. The perturbation expansion was performed in the original t_{2g} basis. The average sign is about 0.48.

8. Summary

We have presented an open-source C++ implementation of the continuous-time hybridization expansion Monte Carlo method for multi-orbital impurity models with general instantaneous two-body interactions and complex hybridization functions. We have discussed the technical details of the implementation, and presented some examples of Monte Carlo simulation results for a three-orbital model, which can serve as a benchmark or reference.

Acknowledgments

We gratefully acknowledge support by the wider ALPS community [30,31]. HS and PW acknowledge support from the Deutsche Forschungsgemeinschaft via FOR 1346, the SNSF (Swiss National Science Foundation) Grant No. 200021E-149122, ERC Advanced Grant SIMCOFE and NCCR MARVEL. We thank Markus Wallerberger and Florian Sohn for useful comments on the manuscript and the code. This work was supported by JSPS KAKENHI Grant Number 15H05885 (J-Physics), 16K17735. EG was supported by DOE ER 46932. Part of the calculations was performed on the ISSP supercomputing system.

Appendix A. Estimation of volumes of configuration spaces

We give a brief description on how to estimate the volumes of configuration spaces in the worm sampling using the Wang-Landau algorithm [18,19]. Let us consider the case of measuring N observables $\mathcal{O}_1, \dots, \mathcal{O}_N$ by the worm sampling. The corresponding configuration spaces are $\mathcal{C}_{\mathcal{O}_1}, \dots, \mathcal{C}_{\mathcal{O}_N}$, while that of the partition function is \mathcal{C}_Z .

We design our Monte Carlo dynamics so that we spend a roughly equal number of steps in each of the $N + 1$ subspaces. This is done by choosing $\eta_i = V_Z/V_i$ in the measurement process, where V_i is the volume of subspace i ($i = \mathcal{O}_1, \dots, \mathcal{O}_N$).

During the thermalization processes, we estimate the subspace volumes following the standard procedure of the Wang-Landau algorithm. In practice, we start a Monte Carlo simulation with an initial guess $V_j = 1$ ($j = Z, \dots, \mathcal{O}_N$). The acceptance rate of a worm insertion/removal is computed using the current values of the weights as $\eta_i = V_Z/V_i$ ($i = \mathcal{O}_1, \dots, \mathcal{O}_N$). After each attempt of a worm insertion or removal, we update V_j as $V_j \rightarrow \lambda V_j$, where j is the current subspace and λ (> 1) is a modification factor. This forces the configuration to visit all the subspaces. During this

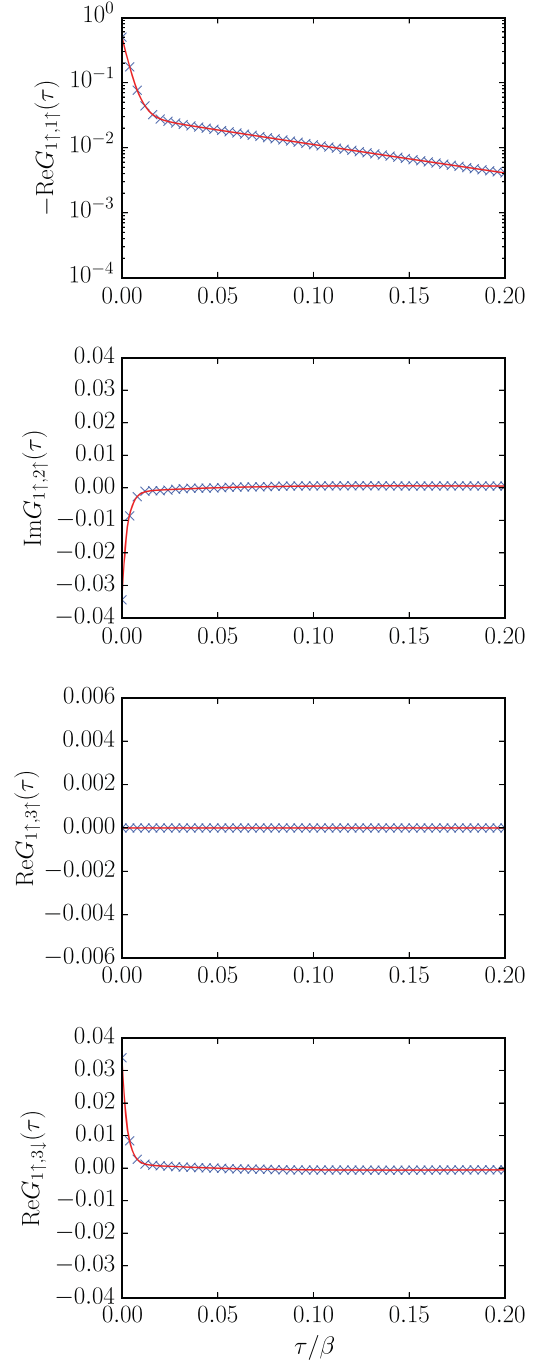


Fig. 2. The single-particle Green's function $G(\tau)$ computed for $\zeta = 1, U = 10, J/U = 1/4, \beta = 40$, and $t' = 0$. The crosses show the results of the CT-HYB code, while the solid lines correspond to the exact results.

random walk between the subspaces, we count the number of Monte Carlo steps N_j spent in each subspace ($j = Z, \mathcal{O}_1, \dots, \mathcal{O}_N$). This random walk is performed until the histogram $\{N_j\}$ becomes reasonably flat. We found that a maximum deviation of 20% from the mean value is a reasonable criterion. Once this criterion is met, we reset the histogram to zero, and update the modification factor as $\lambda \leftarrow \sqrt{\lambda}$. A new random walk is performed until a flat histogram $\{N_j\}$ is again obtained with the new (smaller) modification factor. We repeat this procedure until $\{V_j\}$ ($j = Z, \mathcal{O}_1, \dots, \mathcal{O}_N$) converges within a reasonable accuracy as λ converges to 1.

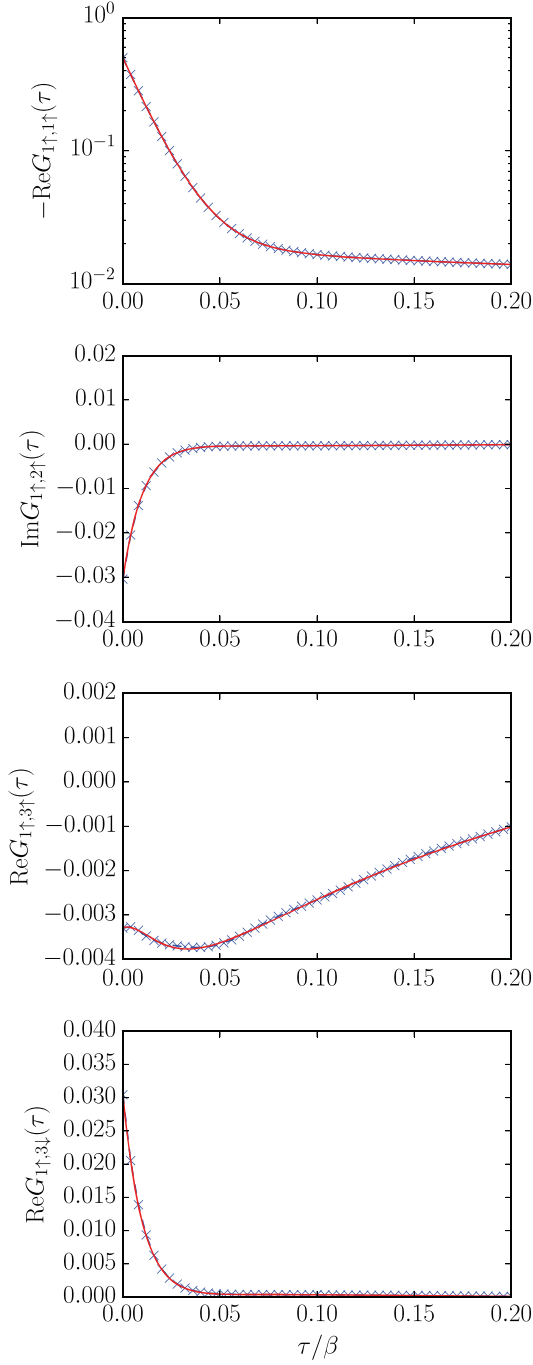


Fig. 3. The single-particle Green's function $G(\tau)$ computed for $\zeta = 1$, $U = 10$, $J/U = 1/4$, $\beta = 10$, and $t' = 0.2$. The crosses show the results of the CT-HYB code, while the solid lines correspond to the exact results.

Appendix B. Auto partitioning of the local Hilbert space

We illustrate our procedure for the auto partitioning of the Hilbert space in Fig. B.4. First, we compute the matrix elements of \mathcal{H}_{loc} in the occupation basis for a given single-particle basis. In the first step [Fig. B.4(a)], we partition the Hilbert space into subspaces so that \mathcal{H}_{loc} becomes a block diagonal matrix. In the second step [Fig. B.4(b)], we unify some of the sectors so that the block matrices of creation and annihilation operators have only a single nonzero block for each row.

If there is no non-zero element, i.e., $\mathcal{H}_{\text{loc}} = 0$, each vector in the occupation basis forms its own subspace (sector). Hereafter,

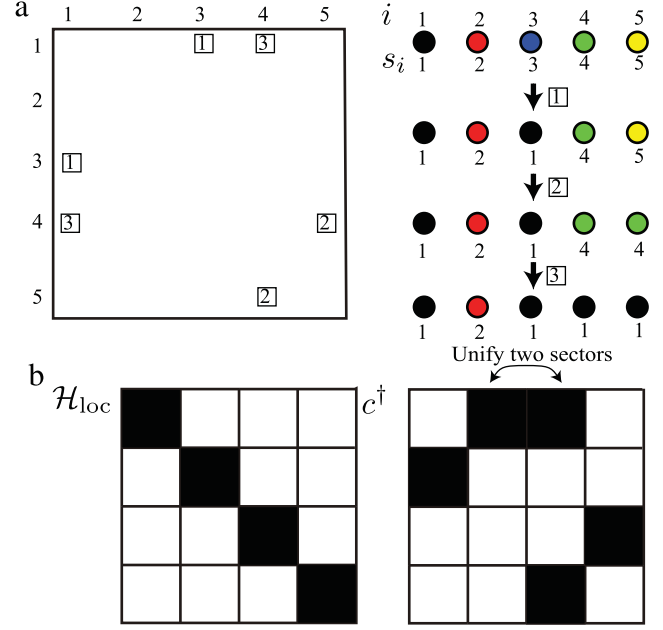


Fig. B.4. (Color online) (a) Partitioning of a 5×5 Hamiltonian. The i th element in the occupation basis belongs to the sector s_i . Sectors are merged by adding off-diagonal elements to \mathcal{H}_{loc} , one by one (squares with numbers). (b) Partitioning of the creation operator.

we denote the sector to which the i th element of the occupation basis belongs by s_i . We add non-zero elements into the matrix of \mathcal{H}_{loc} one by one. Each time we add a new nonzero element $H_{ij} \neq 0$, we unify the two sectors s_i and s_j , if those two vectors belong to different sectors ($s_i \neq s_j$). The procedure is illustrated in Fig. B.4(a) for a 5×5 matrix. After going through all the nonzero elements and reordering rows and columns appropriately, the matrix \mathcal{H}_{loc} becomes block-diagonal.

In the second step [Fig. B.4(b)], we compute the elements of the block matrices of creation operators. If more than two blocks are nonzero in a given row, we unify the corresponding two sectors. This procedure is repeated for each column as well.

The elementary operation of this procedure is unifying two sectors s and s' for given s and s' ($s < s'$). A naive procedure would be to scan through all vectors having s' and assign s to them ($s_i \leftarrow s$ if $s_i = s'$). But this is computationally inefficient for a large number of orbitals because we have to scan the whole Hilbert space many times during the partitioning of the matrices. Instead, we adopt the Hoshen-Kopelman single-pass cluster identification algorithm [32].

Appendix C. Measurement of single-particle Green's function by reconnecting hybridization lines

Here we describe how to generate multiple configurations for the measurement of the single-particle Green's function. First, we present a general procedure for measuring multiple samples from a single configuration c_0 in the worm space \mathcal{C}_A for an observable A . We assume that the thermodynamic average of A is given by

$$\langle A \rangle = \langle f(c_A) \rangle_{\text{MC}}, \quad (\text{C.1})$$

where $\langle \dots \rangle_{\text{MC}}$ is the Monte Carlo average, and f is some function of a configuration in \mathcal{C}_A .

At the measurement step, we randomly generate a set of N configurations, $S = \{c_0, c_1, c_2, \dots, c_{N-1}\}$, including the current one c_0 ($N > 1$). $P_{c \rightarrow S}$ denotes the probability that the set S is

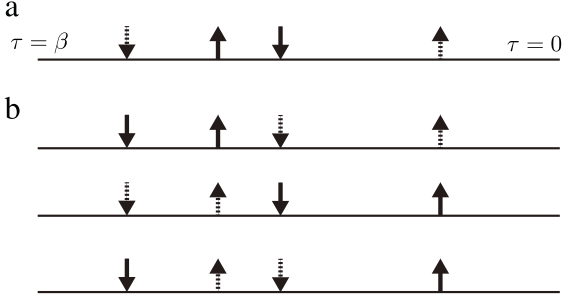


Fig. C.5. (a) Illustration of a configuration with a worm for the single-particle Green's function. Up and down arrows with solid lines represent annihilation and creation operators coupled with the bath, respectively. Annihilation and creation operators of the worm are denoted by dashed arrows. (b) Three configurations can be generated by reconnecting hybridization lines from the configuration in (a).

generated when the current configuration is c . Here, we require $P_{c_i \rightarrow s} = P_{c_j \rightarrow s}$ for $0 \leq i \neq j \leq N-1$. Then, it is easy to prove that

$$\langle A \rangle = \left\langle \frac{\sum_i w_i f(c_i)}{\sum_i w_i} \right\rangle_{MC}. \quad (C.2)$$

Let us consider a configuration in the $G^{(1)}$ space illustrated in Fig. C.5(a). In this case, we can generate three additional configurations shown in Fig. C.5(b) by reconnecting hybridization lines, which does not require the reevaluation of the trace.

We compute the relative Monte Carlo weights of these configurations as follows. (i) First, we attach hybridization lines to the creation and annihilation operators of the worm. (ii) Then, we detach a creation operator and an annihilation operator from the bath. It should be noted that, to avoid a singular matrix, we need to introduce auxiliary fields, which will be removed again in step (ii).

For step (i), we define the following matrix of hybridization functions,

$$\mathbf{M}' = \begin{bmatrix} & & & \Delta_{a'_1, b} & 0 \\ & & & \vdots & \vdots \\ & \mathbf{M}_0 & & \Delta_{a'_N, b} & 0 \\ \Delta_{a, a_1} & \cdots & \Delta_{a, a_N} & \Delta_{a, b} & s_1 \\ 0 & \cdots & 0 & s_2 & s_3 \end{bmatrix}, \quad (C.3)$$

where \mathbf{M}_0 is the matrix defined in Eq. (10) for the current configuration, e.g., the one shown in Fig. C.5(a). The auxiliary fields s_1 , s_2 , and s_3 in the last column and row were introduced to avoid a singular matrix, i.e., $\det \mathbf{M}' = 0$ for $\Delta_{a, b} = 0$. We compute the inverse matrix $(\mathbf{M}')^{-1}$ and the ratio $\det \mathbf{M}' / \det \mathbf{M}_0$ from \mathbf{M}_0^{-1} , $\Delta_{a'_i, b}$, and Δ_{a, a_j} using the fast update formula [14]. The choice of the auxiliary fields may be arbitrary as long as a singular matrix is avoided, because the last column and row will be removed. In practice, we adopt $s_i = \pm \delta$, with δ being a small number (typically around 10^{-5}).

Once $(\mathbf{M}')^{-1}$ is computed, we are ready to compute the relative weight w_{ij} of all the configurations generated by reconnecting hybridization lines, where i and j represent creation and annihilation operators detached from the bath. Counting sign changes from permutations of columns and rows, the relative weight is obtained as

$$\Delta w_{ij} \equiv w_{ij} / w_{nn} = (\det \mathbf{M}' / \det \mathbf{M}_0) \times \det \begin{pmatrix} (\mathbf{M}'^{-1})_{i, j} & (\mathbf{M}'^{-1})_{i, n+1} \\ (\mathbf{M}'^{-1})_{n+1, j} & (\mathbf{M}'^{-1})_{n+1, n+1} \end{pmatrix}, \quad (C.4)$$

where n is the perturbation order. Here, we removed the last column and row as well to avoid systematic errors from the auxiliary fields. In practice, the Monte Carlo average in Eq. (23) is replaced

by the weighted averaged over the configurations

$$\left\langle \frac{\sum_{ij} \Delta w_{ij} \text{sign}(w_{nn}) \text{sign}(\Delta w_{ij}) \tilde{P}_l(\chi(\tau_j - \tau'_i))}{\sum_{ij} |w_{ij} / w_{nn}|} \right\rangle_{G^{(1)}}, \quad (C.5)$$

where $\text{sign}(w_{nn})$ is the sign of the Monte Carlo weight of the current configuration.

Appendix D. Measurement of the two-particle green's function

We measure the two-particle Green's function in the mixed basis representation proposed in Ref. [20]:

$$G_{abcd}^{(2)}(\tau_{12}, \tau_{34}, \tau_{14}) = \sum_{l, l' \geq 0} \sum_{m \in \mathbb{Z}} \frac{\sqrt{2l+1} \sqrt{2l'+1}}{\beta^3} (-1)^{l'+1} \times P_l(\chi(\tau_{12})) P_{l'}(\chi(\tau_{34})) e^{i\omega_m \tau_{14}} G_{abcd}^{(2)}(l, l', i\omega_n), \quad (D.1)$$

where $\omega_n = 2m\pi / \beta$.

A worm for the two-particle Green's function consists of four operators $c_a(\tau_a) c_b^\dagger(\tau'_b) c_c(\tau_c) c_d^\dagger(\tau'_d)$ with four independent flavor and time variables. In the mixed basis representation, the worm estimator reads

$$G_{abcd}^{(2)}(l, l', i\omega_n) = \mathcal{N} \frac{\sqrt{2l+1} \sqrt{2l'+1}}{\beta} (-1)^{l'+1} \times \langle (M_{ab} M_{cd} - M_{ad} M_{cb}) \tilde{P}_l(\tau_a - \tau'_b) \tilde{P}_{l'}(\tau_c - \tau'_d) e^{i\omega_n(\tau_a - \tau'_d)} \rangle_{G^{(2)}}. \quad (D.2)$$

Here, the normalization factor is given by

$$\mathcal{N} = \frac{N_{G^{(2)}}}{\eta_{G^{(2)}} \langle \text{sign} \rangle N_Z}, \quad (D.3)$$

where $N_{G^{(2)}}$ and N_Z represent the numbers of Monte Carlo steps in the worm space and the partition function space, respectively.

As for the single-particle Green's function, we generate multiple samples following the procedure described below. First, we attach hybridization lines to the creation and annihilation operators of the worm by adding three columns and rows –one of each made up of auxiliary fields –to the hybridization-function matrix. Then, we remove three columns and rows.

We define the matrix of the intermediate state by

$$\mathbf{M}' = \begin{bmatrix} & & & \Delta_{a_1, b} & \Delta_{a_1, d} & 0 \\ & & & \vdots & \vdots & \vdots \\ & \mathbf{M}_0 & & \Delta_{a_N, b} & \Delta_{a_N, d} & 0 \\ \Delta_{a, a'_1} & \cdots & \Delta_{a, a'_N} & \Delta_{a, b} & \Delta_{a, d} & s_1 \\ \Delta_{c, a'_1} & \cdots & \Delta_{c, a'_N} & \Delta_{c, b} & \Delta_{c, d} & s_2 \\ 0 & \cdots & 0 & s_3 & s_4 & s_5 \end{bmatrix}, \quad (D.4)$$

where s_i ($i = 1, \dots, 5$) are auxiliary fields. Now, we define $a'_{N+1} \equiv b$, $a'_{N+2} \equiv d$, $a_{N+1} \equiv a$, and $a_{N+2} \equiv c$. Similarly to Eq. (C.4), the relative weight of the worm $c_{a_\alpha} c_{a'_\beta}^\dagger c_{a_\gamma} c_{a'_\eta}^\dagger$ is given by

$$\Delta w_{\alpha\beta\gamma\eta} = (\det \mathbf{M}' / \det \mathbf{M}_0) \times \det(\mathbf{M}'^{-1}; \alpha, \gamma, n+1; \beta, \eta, n+1), \quad (D.5)$$

where we use the shorthand notation

$$\det(\mathbf{A}; i_1 i_2 i_3 : j_1 j_2 j_3) = \det \begin{pmatrix} A_{i_1 j_1} & A_{i_1 j_2} & A_{i_1 j_3} \\ A_{i_2 j_1} & A_{i_2 j_2} & A_{i_2 j_3} \\ A_{i_3 j_1} & A_{i_3 j_2} & A_{i_3 j_3} \end{pmatrix}. \quad (D.6)$$

We omitted the index for imaginary time to simplify the notation. In contrast to the measurement of the single-particle Green's function, we do not take the summation of α , β , γ , and η over all columns and rows, because it would cost $O(n^4)$, which is more

expensive than the Monte Carlo updates [$O(n^3)$]. Instead, we generate two sets S_{col} and S_{row} by selecting a few elements from $1, 2, \dots, n+1, n+2$ so that they always include $n+1$ and $n+2$. The size of the two sets n' is taken to be typically around 5–10. The summation is taken over these two sets as $\alpha, \gamma \in S_{\text{col}}$ and $\beta, \eta \in S_{\text{row}}$.

References

- [1] A. Georges, G. Kotliar, W. Krauth, M.J. Rozenberg, Rev. Modern Phys. 68 (1) (1996) 13. <http://dx.doi.org/10.1103/RevModPhys.68.13>.
- [2] T. Maier, M. Jarrell, T. Pruschke, M.H. Hettler, Rev. Modern Phys. 77 (3) (2005) 1027–1080. <http://dx.doi.org/10.1103/RevModPhys.77.1027>. URL <http://link.aps.org/doi/10.1103/RevModPhys.77.1027>.
- [3] G. Kotliar, S. Savrasov, K. Haule, V. Oudovenko, O. Parcollet, C. Marianetti, Rev. Modern Phys. 78 (3) (2006) 865–951. <http://dx.doi.org/10.1103/RevModPhys.78.865>. URL <http://link.aps.org/doi/10.1103/RevModPhys.78.865>.
- [4] H. Shinaoka, S. Hoshino, M. Troyer, P. Werner, Phys. Rev. Lett. 115 (2015) 156401.
- [5] A. Toschi, A. Katanin, K. Held, Phys. Rev. B 75 (4) (2007). <http://dx.doi.org/10.1103/PhysRevB.75.045118>. URL <http://link.aps.org/doi/10.1103/PhysRevB.75.045118>, pp. 045118–8.
- [6] A.N. Rubtsov, M.I. Katsnelson, A.I. Lichtenstein, Phys. Rev. B 77 (3) (2008). <http://dx.doi.org/10.1103/PhysRevB.77.033101>. URL <http://link.aps.org/doi/10.1103/PhysRevB.77.033101>, pp. 033101–4.
- [7] A. Rubtsov, V. Savkin, A. Lichtenstein, Phys. Rev. B 72 (3) (2005) 035122. <http://dx.doi.org/10.1103/PhysRevB.72.035122>. URL <http://link.aps.org/doi/10.1103/PhysRevB.72.035122>.
- [8] P. Werner, A. Comanac, L. de' Medici, M. Troyer, A. Millis, Phys. Rev. Lett. 97 (7) (2006) 076405. <http://dx.doi.org/10.1103/PhysRevLett.97.076405>. URL <http://link.aps.org/doi/10.1103/PhysRevLett.97.076405>.
- [9] E. Gull, P. Werner, A. Millis, M. Troyer, Phys. Rev. B 76 (23) (2007) 235123. <http://dx.doi.org/10.1103/PhysRevB.76.235123>. URL <http://link.aps.org/doi/10.1103/PhysRevB.76.235123>.
- [10] H. Hafermann, P. Werner, E. Gull, Comput. Phys. Comm. 184 (4) (2013) 1280–1286. <http://dx.doi.org/10.1016/j.cpc.2012.12.013>.
- [11] P. Seth, I. Krivenko, M. Ferrero, O. Parcollet, Comput. Phys. URL <http://www.sciencedirect.com/science/article/pii/S001046551500404X4>.
- [12] L. Huang, Y. Wang, Z.Y. Meng, L. Du, P. Werner, X. Dai, Comput. Phys. Comm. 195 (2015) 140–160. <http://dx.doi.org/10.1016/j.cpc.2015.04.020>.
- [13] N.V. Prokof'ev, B.V. Svistunov, I.S. Tupitsyn, Phys. Lett. A 238 (4–5) (1998) 253–257. [http://dx.doi.org/10.1016/S0375-9601\(97\)00957-2](http://dx.doi.org/10.1016/S0375-9601(97)00957-2). URL <http://linkinghub.elsevier.com/retrieve/pii/S0375960197009572>.
- [14] E. Gull, A.J. Millis, A.I. Lichtenstein, A.N. Rubtsov, M. Troyer, P. Werner, Rev. Modern Phys. 83 (2) (2011) 349–404. <http://dx.doi.org/10.1103/RevModPhys.83.349>. URL <http://link.aps.org/doi/10.1103/RevModPhys.83.349>.
- [15] P. Gunacker, M. Wallerberger, E. Gull, A. Hausoel, G. Sangiovanni, K. Held, Phys. Rev. B 92 (15) (2015) 155102–10. <http://dx.doi.org/10.1103/PhysRevB.92.155102>. URL <http://link.aps.org/doi/10.1103/PhysRevB.92.155102>.
- [16] P. Gunacker, M. Wallerberger, T. Ribic, A. Hausoel, G. Sangiovanni, K. Held, Phys. Rev. B 94, 125153. URL <https://doi.org/10.1103/PhysRevB.94.125153>.
- [17] P. Werner, A. Millis, Phys. Rev. B 74 (15) (2006) 155107. <http://dx.doi.org/10.1103/PhysRevB.74.155107>. URL <http://link.aps.org/doi/10.1103/PhysRevB.74.155107>.
- [18] F. Wang, D.P. Landau, Phys. Rev. Lett. 86 (10) (2001) 2050–2053. <http://dx.doi.org/10.1103/PhysRevLett.86.2050>. URL <http://link.aps.org/doi/10.1103/PhysRevLett.86.2050>.
- [19] F. Wang and D.P. Landau, Phys. Rev. E, 64, 056101. <https://doi.org/10.1103/PhysRevE.64.056101>.
- [20] L. Boehnke, H. Hafermann, M. Ferrero, F. Lechermann, O. Parcollet, Phys. Rev. B 84 (7) (2011) 075145. <http://dx.doi.org/10.1103/PhysRevB.84.075145>. URL <http://link.aps.org/doi/10.1103/PhysRevB.84.075145>.
- [21] K. Haule, Phys. Rev. B 75 (15) (2007) 155113. <http://dx.doi.org/10.1103/PhysRevB.75.155113>. URL <http://link.aps.org/doi/10.1103/PhysRevB.75.155113>.
- [22] N. Parragh, A. Toschi, K. Held, G. Sangiovanni, Phys. Rev. B 15 (2012) 155158. <http://dx.doi.org/10.1103/PhysRevB.86.155158>. arXiv:1209.0915v1. URL <http://link.aps.org/doi/10.1103/PhysRevB.86.155158>.
- [23] N. Parragh, Strongly correlated multi orbital systems—a continuous-time quantum Monte Carlo analysis (Ph.D. thesis), University of Würzburg, 2013. URL <https://opus.bibliothek.uni-wuerzburg.de/frontdoor/index/index/docId/7334>.
- [24] H. Shinaoka, M. Dolfi, M. Troyer, P. Werner, J. Stat. Mech. Theory Exp. 2014 (6) (2014) P06012. <http://dx.doi.org/10.1088/1742-5468/2014/06/P06012>. URL <http://stacks.iop.org/1742-5468/2014/i=6/a=P06012?key=crossref.371464d32ba9ab9f813d517b81ad778e>.
- [25] A.M. Läuchli, P. Werner, Phys. Rev. B 80 (23) (2009) 235117. <http://dx.doi.org/10.1103/PhysRevB.80.235117>. URL <http://link.aps.org/doi/10.1103/PhysRevB.80.235117>.
- [26] P. Sémon, A.M.S. Tremblay, Phys. Rev. B 85 (20) (2012) 201101–5. <http://dx.doi.org/10.1103/PhysRevB.85.201101>. URL <http://link.aps.org/doi/10.1103/PhysRevB.85.201101>.
- [27] H. Shinaoka, Y. Nomura, S. Biermann, M. Troyer, P. Werner, Phys. Rev. B 92 (19) (2015) 195126–14. <http://dx.doi.org/10.1103/PhysRevB.92.195126>. URL <http://link.aps.org/doi/10.1103/PhysRevB.92.195126>.
- [28] A. Gaenko, A.E. Antipov, G. Carcassi, T. Chen, X. Chen, Q. Dong, L. Gamper, J. Gukelberger, R. Igarashi, S. Isakov, M. Könz, J.P.F. LeBlanc, R. Levy, P.N. Ma, J.E. Paki, H. Shinaoka, S. Todo, M. Troyer, E. Gull, Comput. Phys. Comm. 213 (2017) 235–251. <http://dx.doi.org/10.1016/j.cpc.2016.12.009>. URL [arXiv:1609.03930v1](https://arxiv.org/abs/1609.03930v1).
- [29] A.E. Antipov, I. Krivenko, pomerol 1.1 [online] (May 2015). <http://dx.doi.org/10.5281/zenodo.17900>.
- [30] B. Bauer, L.D. Carr, H.G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, J. Stat. Mech. Theory Exp. (05) (2011) P05001. URL <http://iopscience.iop.org/1742-5468/2011/05/P05001>.
- [31] A.F. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gürtler, A. Honecker, R. Igarashi, M. Körner, A. Kozhevnikov, A. Läuchli, S.R. Manmana, M. Matsumoto, I.P. Mcculloch, F. Michel, R.M. Noack, G. Pawłowski, L. Pollet, T. Pruschke, U. Schollwock, S. Todo, S. Trebst, M. Troyer, P. Werner, S. Wessel, J. Magn. Magn. Mater. 310 (2) (2007) 1187–1193. <http://dx.doi.org/10.1016/j.jmmm.2006.10.304>. URL <http://linkinghub.elsevier.com/retrieve/pii/S0304885306014983>.
- [32] J. Hoshen, R. Kopelman, Phys. Rev. B. (2017). URL <http://journals.aps.org/prb/abstract/10.1103/PhysRevB.14.3438>.